

## FUNGSI

Blok fungsi hampir sama dengan blok prosedur, hanya fungsi harus dideklarasikan dengan tipenya. Tipe deklarasi ini menunjukkan tipe hasil dari fungsi.

Bentuk umum :

FUNCTION identifier (daftar parameter) : type ;

Blok fungsi juga diawali dengan kata cadangan Begin dan di akhiri dengan kata cadangan End dan titik koma.

Perbedaan fungsi dengan prosedur adalah :

- Pada fungsi, nilai yang dikirimkan balik terdapat pada nama fungsinya ( kalau pada prosedur pada parameter yang dikirimkan secara acuan).
- Karena nilai balik berada di nama fungsi tersebut, maka fungsi tersebut dapat langsung digunakan untuk dicetak hasilnya. Atau nilai fungsi tersebut dapat juga langsung dipindahkan ke pengenalan variable yang lainnya.
- Pada prosedur, nama prosedur tidak dapat digunakan langsung, yang dapat langsung digunakan adalah parameternya yang mengandung nilai balik.

### 1.1 Fungsi Tanpa Parameter

Fungsi yang tanpa parameter berarti nilai balik yang akan dihasilkan merupakan nilai yang sudah pasti. Karena tidak menggunakan parameter, maka hasil fungsi tersebut tidak dapat diatur dari modul yang menggunakannya, karena tidak ada parameter yang dikirimkan. Fungsi tanpa parameter jarang dipergunakan.

Contoh program :

```
Type Huruf = string [6] ;  
Function Garis : Huruf ;  
Begin  
    Garis := '=====';  
End ;  
Begin  
    Writeln (Garis) ; Writeln ('Pascal') ; Writeln (Garis) ;  
End.
```

Output program :

```
=====
Pascal
=====
```

Keterangan program :

Fungsi “Garis” hanya menghasilkan nilai yang sudah pasti yaitu :  
“=====”.

### 1.2 Parameter Dalam Fungsi

Parameter dalam fungsi dapat dikirimkan secara nilai atau secara acuan.

Contoh penulisan parameter dengan pengiriman secara nilai :

```
Function Hitung (A, B : integer ) : integer ;
```

Contoh program :

```
Function Terbesar(X, Y : real ) : real ;
Begin
    If X > Y Then
        Terbesar := X
    Else
        Terbesar := Y ;
End ;
Var
    Nilai1 , Nilai2 : real ;
Begin
    Write ('Nilai pertama =') ; readln (Nilai1) ;
    Write ('Nilai kedua = ') ; readln (Nilai2) ;
    Writeln ('Nilai terbesar adalah =', Terbesar (Nilai1, Nilai2) : 9 : 3 ) ;
End.
```

Output program :

```
Nilai pertama = 12.356
Nilai kedua = 55.182
Nilai terbesar adalah = 55.182
```

Penulisan judul fungsi yang menggunakan parameter dengan pengiriman secara acuan adalah dengan menambahkan kata cadangan var.

Contoh penulisan :

Function Hitung ( Var A, B : integer ) : integer ;

Pengiriman parameter secara acuan akan mengakibatkan perubahan nilai parameter di fungsi juga merubah nilai parameter di modul yang mengirimkannya. Fungsi yang menggunakan pengiriman parameter secara acuan ini mirip dengan prosedur, yaitu parameter yang dikirimkan secara acuan tersebut dapat di manfaatkan sebagai hasil balik.

### 1.3 Fungsi Pangkat

Pascal tidak menyediakan fungsi untuk perpangkatan tinggi, yang ada hanya fungsi standar Sqr, yaitu pemngkatan kuadrat saja. Bila akan melakukan perpangkatan lebih dari pangkat dua, maka harus dibuat program tersendiri.

### 1.4 Fungsi Memanggil Dirinya Sendiri

Proses fungsi memanggil dirinya sendiri juga merupakan proses recursion.

Contoh program :

```
Function Faktorial (Nilai : integer ) : real ;
Begin
    If Nilai = 0 Then
        Faktorial := 1
    Else
        Faktorial := Nilai * Faktorial (Nilai - 1 ) ;
    End ;
Var
    N : integer ;
Begin
    Write ('Berapa factorial ?') ;
    Readln (N) ;
    Writeln (N, 'faktorial =', Faktorial(N) : 9 : 0 ) ;
End.
```

Output program :

Berapa factorial ? 5  
5 faktorial = bbbbbb120

ket : b = blank

### **1.5 Fungsi Memanggil Fungsi Yang Lain**

Fungsi yang di panggil letaknya harus berada di atas fungsi yang memanggilnya.

Contoh program :

```
Function Fungsi2 (Y : integer) : integer ;  
Begin  
    Fungsi2 := Y * 2 ;  
End ;  
Function Fungsi1 ( X : integer ) : integer ;  
Begin  
    Fungsi1 := Fungsi2 (X) + 5 ;  
End ;  
Begin  
    Writeln (Fungsi1(3)) ;  
End.
```

### **1.6 Acuan Forward Pada Fungsi**

Fungsi dapat dideklarasikan terpisah dengan bloknya. Deklarasi fungsi yang terpisah dengan blok fungsinya dipergunakan kata cadangan Forward dan blok fungsinya juga di deklarasikan ulang, tetapi tidak perlu menyebutkan parameternya.

### **1.7 Fungsi Tersarang**

Merupakan fungsi yang letaknya berada di dalam fungsi yang lainnya.

Contoh program :

```
Function Fungsi1 (X : integer ) : integer ;  
    Function Fungsi2 ( Y : integer ) : integer ;  
        Begin  
            Fungsi2 := Y * 2 ;  
        End ;  
Begin  
    Fungsi1 := Fungsi2 (X) + 5 ;  
End ;  
Begin  
    Writeln (Fungsi1 (3)) ;  
End.
```

## 1.8 Fungsi Standar

### 1.8.1 Fungsi Standar Aritmatika

Ada Abs, ArcTan, Cos, Exp, Franc, Int, Ln, Pi, Sin, Sqr dan Sqrt

Penjelasan :

- Abs (x) ;  
Memutlakkan suatu nilai yang ditunjukkan oleh argument x. dapat berupa tipe real atau integer.
- Exp (x : real ) : real ;  
Untuk menghitung nilai pangkat dari bilangan e. dapat berupa real atau integer dan hasil dari fungsinya adalah real.
- Ln (x : real ) : real ;  
Untuk menghitung nilai logaritma alam dari nilai x. tipe datanya real atau integer dan hasil fungsinya adalah real.
- Int (x : real) : real ;  
Untuk menghasilkan nilai integer dari x. hasil dari fungsi adalah tipe real dengan nilai yang berupa pembulatan ke bawah dari nilai x.
- Franc (x : real ) : real ;  
Untuk mendapatkan nilai pecahan dari argument x. Argumen x dapat berbentuk real atau integer dan hasilnya adalah real.
- Sqr (x) ;  
Untuk menghitung nilai pangkat kuadrat dari argument x.
- Sqrt (x : real ) : real ;  
Untuk menghitung nilai akar dari argument x. dapat real dan integer dan hasil dari fungsinya adalah real.
- Pi  
Akan menghasilkan nilai PI sebesar 3, 14.

### 1.8.2 Fungsi Standar Transfer

Digunakan untuk merubah suatu nilai ke bentuk nilai yang lain. Yang tersedia adalah fungsi standar Chr, Ord, Round dan trunc.

Penjelasan :

- Chr (x : byte) : char ;  
Untuk merubah nilai dari byte x ke bentuk karakter yang sesuai dengan kode ASCII.
- Ord (x) : longint ;  
Untuk merubah nilai x ke bentuk nilai longint yang sesuai dengan kode ASCII. Fungsi ini kebalikan dari fungsi standar Chr.
- Round (x : real) : longint ;  
Untuk membulatkan nilai dari real x ke nilai longint yang terdekat. Bila nilai pecahan sama dengan atau lebih besar dari 0,5 akan dibulatkan ke atas, sedangkan bilai nilai pecahan lebih kecil dari 0,5 akan dibulatkan ke bawah.
- Trunc (x : real) : longint ;  
Untuk membulatkan nilai dari real x ke nilai longint terkecil, atau dengan kata lain membulatkan ke bawah.

### 1.8.3 Fungsi Standar Lainnya

Terdiri dari : Hi, Lo, Random, SizeOf, Swap, UpCase, ParamCount dan ParamStr.

Penjelasan :

- Hi (x) : byte ;  
Untuk mengisi low order byte dari hasil fungsi dengan high order byte dari ungkapan integer x. high order byte dari hasil fungsi akan bernilai nol. Tipe hasil dari fungsi ini adalah byte.
- Lo (x) : byte ;  
Untuk mengisi low order byte dari hasil fungsi dengan low order byte dari ungkapan integer x. high order byte dari hasil fungsi akan bernilai nol. Tipe hasil dari fungsi ini adalah byte.
- Swap (x) ;  
Untuk membalik bit-bit di low order byte menjadi high order byte dan sebaliknya dari ungkapan x. ungkapan x dapat berupa tipe integer atau word.
- Random [ (range : word ) ] ;  
Untuk menghasilkan angka random berkisar dari nilai lebih besar atau sama dengan nol dan lebih kecil dari satu. Bila range tidak disebutkan, hasil dari fungsi ini adalah real. Bila range disebutkan, hasilnya adalah word.
- SizeOf (x) : word ;  
Untuk menunjukkan besarnya byte yang digunakan oleh suatu variable x. hasilnya adalah word.

- UpCase (Ch : char ) : char ;  
Untuk merubah argument suatu karakter yang ditunjukkan oleh Ch menjadi bentuk karakter huruf besar ( upper case ).
- ParamCount: word ;  
Untuk mengetahui jumlah parameter yang dikirimkan lewat prompt DOS
- ParamStr (index) : string ;  
Untuk menerima parameter yang dikirimkan lewat prompt DOS, index adalah ungkapan dengan tipe word, dan akan menghasilkan parameter ke index yang dikirimkan tersebut.